1.0

1.1

1.25   1.4   1.6

.8   .25

2.2

2.0

1.8

AD A110382

LEVEL II ⑫

TACTICAL AIR SURVEILLANCE RADAR NETTING (TASRAN) SIMULATOR/EMULATOR

General Research Corporation

G. F. Van Blaricum

DTIC
SELECTE
FEB 0 3 1982
E

DTIC FILE COPY

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

82 02 02 047

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-81-278 has been reviewed and is approved for publication.

APPROVED: *George A. Ellis*

GEORGE A. ELLIS
Project Engineer

APPROVED: *Frank J. Rehm*

FRANK J. REHM
Technical Director
Surveillance Division

FOR THE COMMANDER: *John P. Huss*

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (OCTM) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
RADC-TR-81-278	AD-A110 382	

4. TITLE (and Subtitle)	5. TYPE OF REPORT & PERIOD COVERED
TACTICAL AIR SURVEILLANCE RADAR NETTING (TASRAN) SIMULATOR/EMULATOR	Final Technical Report March 1980 - June 1981
	6. PERFORMING ORG. REPORT NUMBER
	GRC-CR-6-994

7. AUTHOR(s)	8. CONTRACT OR GRANT NUMBER(s)
G. F. Van Blaricum	F30602-80-C-0154

9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
General Research Corporation P O Box 6770 Santa Barbara CA 93110	62702F 45061678

11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE
Rome Air Development Center (OCTM) Griffiss AFB NY 13441	October 1981
	13. NUMBER OF PAGES
	48

14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office)	15. SECURITY CLASS. (of this report)
Same	UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
	N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: George A. Ellis (OCTM)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Radar	Simulation
Netting	Tracking
TACS	Communication

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

TASRAN is a computer simulation for evaluating netted tactical air surveillance systems. TASRAN can model user-defined networks of ground-based radars, and it can exercise this radar network against user-defined scenarios of hostile and friendly aircraft on specified flight paths. The data processing functions of radar detection, track initiation and correlation, and automatic target tracking are performed (not merely simulated) on the simulated measurements from the network of radars.

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Communication from site to site can be modeled in various levels of detail, from simple message routing to resource sharing and delays to detailed calculations of propagation and jamming degradation. Simulation outputs can be summarized in printed tables or in interactive displays at a Tektronix graphics terminal.

The computer program is installed on RADC's Honeywell 6180 computer system; an identical version is installed on GRC's DEC VAX-11/780, where all of the code was developed and tested.

Accession For	
NTIS CRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

DTIC
COPY
INSPECTED
1

CONTENTS

ILLUSTRATIONS

1   INTRODUCTION

TASRAN is a computer simulation for evaluating netted tactical air surveillance systems. It can simulate arbitrary types and numbers of radars, aircraft, and communication links. TASRAN can also act as an emulator, in the sense that it will accept actual radar measurements or radar track messages in standard formats and process these along with simulated measurements and simulated tracks.

TASRAN can be used in several ways. It can be used to examine specific radar netting configurations, with user-specified types and numbers of radars, data processors, and communication links and nodes and with a scenario of aircraft, helicopters, and jammers. TASRAN can also be used to experiment with radar data processing software, such as track filters, for any netting configuration. In addition, TASRAN will be a tool for investigating more general netting functions such as jammer triangulation, self-survey, and multi-sensor target identification. However, these latter functions will require some additions to the basic TASRAN software.

TASRAN consists of about 15,000 lines of new source code written in CIFTRAN, a structured version of FORTRAN developed by General Research Corporation (GRC). An additional 20,000 lines of software, written in CIFTRAN, FORTRAN, and PL/1, were supplied with TASRAN to provide support functions such as interactive graphics, CIFTRAN translation, program library maintenance, post-processing, etc.

The computer program is installed on RADC's Honeywell 6180 computer system; an identical version is installed on GRC's DEC VAX-11/780, where all of the code was developed and tested.

1.1   PURPOSE OF THE FINAL REPORT

TASRAN is described in a number of reports listed in Sec. 1.2. This Final Report supplements the other reports on TASRAN. It describes

some of the specific models used in TASRAN; it reports the testing per-
formed on TASRAN before it was delivered to RADC; and it suggests some
areas in which TASRAN could be upgraded to provide some additional capa-
bilities for analysis of netting systems and to reduce its running time.
In addition, Sec. 1.3 of this report provides an overview description of
TASRAN.

1.2 PROJECT REFERENCES *

The references pertinent to this project are:

1. Design Plan for the Tactical Air Surveillance Radar Netting
    Simulator/Emulator, General Research Corporation CR-1-994,
    January 1981.

2. Test Plan for the Tactical Air Surveillance Radar Netting
    Simulator/Emulator, General Research Corporation CR-2-994,
    February 1981.

3. Functional Description for the Tactical Air Surveillance
    Radar Netting Simulator/Emulator, General Research Corpora-
    tion CR-3-994, May 1981.

4. Program Maintenance Manual for the Tactical Air Surveillance
    Radar Netting Simulator/Emulator, General Research Corpora-
    tion CR-4-994, May 1981.

5. User's Manual for the Tactical Air Surveillance Radar Net-
    ting Simulator/Emulator, General Research Corporation
    CR-5-994, May 1981.

1.3 TASRAN CAPABILITIES

TASRAN can model a user-defined network of ground-based radars,
and it can exercise this rada. network against scenarios of large num-
bers of hostile and friendly aircraft on specified flight paths. The
data processing functions of radar detection, track initiation and cor-
relation, and automatic target tracking are performed (not merely simu-
lated) on the simulated measurements from the network of radars. Com-
munication from site to site can be modeled in various levels of detail,

from simple message routing to resource sharing and delays to detailed calculations of propagation and jamming degradation. Simulation outputs can be summarized in printed tables or in interactive displays at a Tektronix graphics terminal.

A summary of the models in TASRAN is provided in Table 1.1.

1.3.1 Radars

Two classes of radars are simulated in TASRAN—a conventional rotating surveillance radar operating in a track-while-scan mode and a stationary phased-array radar operating under computer control. Various types of track-while-scan (TWS) radars (such as the AN/TPS-43E) are defined by such parameters as power, antenna beamwidth and gain, losses, noise figure, and scan rate; types of computer-directed-track (CDT) radars (such as Army's PATRIOT) are specified by these parameters plus array face orientation and surveillance scan pattern. Any number of types of TWS and CDT radars can be defined. Individual radars are specified by their type and by their location (including siting errors in position and orientation). In general, there may be any number of radars of any particular type.

Radars in TASRAN produce simulated measurements of a target's range and angles. Measurements are degraded by pseudo-random noise that depends on parameters of the radar and the signal-to-interference ratio (S/I) of the measurement (noise plus clutter plus jamming). These simulated measurements, along with such parameters as measurement time and S/I, are passed to the data processor associated with the radar.

1.3.2 Data Processors

Each simulated radar has a data processor which processes all of the measurements. Each simulated data processor actually performs a version of track initiation, track correlation, and track updating equivalent to that performed at an actual site. A data processing site is

TABLE 1.1

TASRAN MODEL SUMMARY

Radars
- Rotating Surveillance Radars (e.g., TPS-43E)
- Fixed Phased-Array Radars (e.g., PATRIOT)
- Radar Parameters, such as:
  — Power
  — Antenna Pattern
  — Coverage
  — Clutter Rejection

Radar Data Processors
- Track Initiation
  — 2-Hit and 3-Hit Algorithm
- Track Association
- Track Filter
  — Kalman and Alpha-Beta
- Radar Scheduler
- Track File Maintenance

Control Module Data Processors
- Message Handling
- Track Association
- Track Filter
- Track File Maintenance

Scenario
- Aircraft
  — Great-Circle Flight Segments
  — Pursuit Navigation With Maneuver Limits
- Helicopters
  — Hovering and Vertical Flight
- Jammers
  — Radar Noise Jamming
  — Communication Noise, Tone, and FM Jamming
- Local Terrain Around Radar
  — Range and Elevation of Horizon
  — Clutter Cross Section
  — Discrete Targets

Communication Network
- Configuration
  — Point-to-Point Wire or Radio
  — Broadcast Network (JTIDS)
- Terminal Equipment
  — Transceiver Types
  — Antennas
  — Data Rates/Formats
  — ECCM Features
- Propagation
- Routing and Queuing
  — Primary and Backup Routes
  — Delays for Encoding and Transmission
  — Waits for Busy Equipment

specified by its data processor type, its location, its connection to a
radar, and its connection to the communication network. The data pro-
cessor type defines such features as the type of track initiation algo-
rithm (2-hit or 3-hit), the type of track filter (alpha-beta or Kalman
filter), and some numerical parameters used in the various algorithms.

In addition to the data processors connected to individual radars,
there are data processor sites that function as operation centers.
These sites, which in the context of the current Tactical Air Control
System can be compared to CRPs or CRCs, perform the functions of combin-
ing the many track reports (new tracks, track updates, or dropped
tracks) from other data processors into a consistent, non-duplicated
file of tracks. A track can be updated either by picking the best of
the reports relating to it or by merging all of the track reports using
an optimal weighting scheme. Tracks maintained by these sites may be
passed to higher-level operation centers which can process tracks from
both radar sites and lower-level operation centers.

1.3.3 Communication System

The communication system that ties together the sensors and data
processors in the radar net can be modeled in three different levels of
detail. At the simplest level, the network is specified merely by its
connectivity—that is, who communicates with whom. At this level, mes-
sages are passed from site to site without delay or degradation and
without regard to any sharing of limited communication resources. This
level of communication system modeling is valuable for investigating
characteristics of radar netting that depend primarily on the sensors,
data processors, or scenario. This version also requires the least
analyst time to specify and the least computer time to execute.

The next two levels of communication system simulation employ the
Tactical Communication System (TCS) model developed by GRC for HQ
USAF/SA and adapted to TASRAN. The TCS model requires a detailed defi-
nition of the communication network in terms of the terminal equipment

5

at all nodes, routing from node to node, and primary and backup circuits and combinations of circuits from node to node. The "FAST" version of the TCS model primarily simulates delays in the transmission of messages due to the limited data rate of the link or the terminal equipment or due to busy circuits or equipment. The "TECH" or technical version includes the capabilities of the FAST model plus calculations of message degradation due to propagation losses or jamming. The TECH model requires that the communication network be realistically specified in terms of equipment power, antenna gains, and system losses.

1.3.4 Scenarios

TASRAN scenarios can include arbitrary types and numbers of aircraft, ground-fixed targets, and radar and communication jammers. Three types of aircraft flight models are provided. Aircraft can fly great-circle segments from point to point at constant or gradually changing altitude. They may also fly pursuit-navigation paths through a sequence of points at given speeds. The pursuit-navigation model includes limitations on turn rates and accelerations due to specified aircraft limits, aircraft speed, and air density. A helicopter flight model simulates vertical flight and hovering.

Aircraft and ground targets have a radar cross section that can be modeled as aspect-dependent or not, and as constant or fluctuating according to a Swerling law.

Radar and communication jammers are specified by power, frequency coverage, antenna gain and pointing direction, and by location on the ground or deployment on an aircraft. Radar jammers are assumed to be noise jammers; communication jammers can employ some additional jamming modulations.

1.3.5 Support Software

The support software for TASRAN consists of a GRC simulation system which has been designed, constructed, and improved over the past

several years and used in nearly all of GRC's large-scale computer simulations. This simulation support software provides the essential features required by major dynamic simulations: complete modularity, readable structured code, complete and readable dataset description, easy and flexible input techniques, dynamic storage allocation with complete data management capabilities, an event-based simulation capability that permits such features as delays during a sequence of related events, a program library maintenance system, and a flexible postprocessor that aids in presenting simulation output in useful tabular and graphical form.

The specific elements of support software that are provided with TASRAN are:

- CIFTRAN. A preprocessor that adds structured language statements to FORTRAN and provides hierarchically indented printout.

- DSA. A dynamic storage allocation and data management system.

- PRINTOUT. A program that aids in defining and maintaining a large dataset and file structure.

- FLEXREAD. A program that allows free-form input of datasets by their logical data names.

- GRCSIM. A package of utility routines for performing run-time simulation support.

- POSTPROCESSOR. A program that summarizes and displays output data in a variety of formats.

- UPDATX. A program that provides a convenient means of creating and maintaining a program library.

1.4 SUMMARY OF THE FINAL REPORT

1.4.1 Modeling

Aircraft Flight Models. TASRAN provides models for aircraft
flight along a succession of great-circle paths and for maneuvering
flight through closely spaced points within the maneuver limits of the
aircraft. In addition, a helicopter model provides for hovering and
vertical flight. The flight models are used to build a table of flight
position and velocity vectors at fixed time steps for each aircraft.
The position and velocity at any time are found by quadratic interpola-
tion in this table.

Radar Cross Section. The radar cross section provides for aspect
dependence with each target being described by its nose-on and side-on
average RCS. A target may have a constant RCS or Swerling I variation.

Antenna Pattern. The antenna pattern of each radar is modeled in
two planes (azimuth and elevation for a dish; u and v for an array)
by a model which specifies the peak gain, mainlobe beamwidth, peak side-
lobe level, and backlobe level. The mainlobe is given a sin (x)/x
shape. Sidelobes and backlobes are modeled by their average values;
their peaks and nulls are not modeled.

Noise Modeling. The noise model affects the radar in three ways--
it produces false alarm detections, it affects the detection probability
of actual targets, and it degrades the accuracy of the radar measure-
ments. Each radar is assigned an average false alarm rate. False
alarms occur at times drawn at random from a Poisson distribution nor-
malized to produce the desired average rate. A pseudo-random value is
produced for thermal noise. Total radar interference is modeled as the
sum of this value of thermal noise plus jamming noise and clutter. Then
the target signal-to-interference ratio is compared to a fixed threshold
to determine whether or not the target is detected. Radar measurements

are corrupted by pseudo-random Gaussian noise to model the effects of random and bias errors in measurements of range and angles.

Track Initiation. Two track initiation algorithms are provided in TASRAN. The "two-hit" algorithm uses two successive measurements to estimate a target's velocity. If the velocity is credible, and if the target does not associate wtih a known track, a new track is established. The "three-hit" algorithm uses two measurements to establish a "trial" track, and then a third measurement to confirm it.

Track Filter. TASRAN provides two different track filters--a fully coupled six-state Kalman filter, and an alpha-beta filter. The alpha-beta filter includes maneuver detection to change the filter weights on a maneuvering target.

Phased Array Models. Two special models are used for phased array radars. The radar operates under pulse-by-pulse computer control, and each radar pulse is individually scheduled. The radar scheduler uses the following priority scheme in choosing which pulse to schedule next: (1) track initiation, (2) track, and (3) surveillance. The surveillance pulses are generated to follow a search raster defined by inputs to TASRAN. The search raster consists of a number of constant-elevation sweeps in fixed azimuth steps between defined azimuth limits. The radar scans between specified range limits on each sweep.

1.4.2 TASRAN Testing

Before it was delivered to RADC, TASRAN underwent three levels of testing. These are (1) development testing, (2) RXVP80 testing, and (3) demonstation testing. The tests used are described more fully in the TASRAN Test Plan.

Development testing consisted of testing the TASRAN models as they were being developed. Many of the models were tested outside of the main simulation using specially generated test programs.

9

RXVP80 testing used the RXVP80 program validation tools developed by GRC for testing FORTRAN programs. This testing proceeded in two steps.

The first step was "static analysis," which examined all of the simulation code for errors in subroutine and function calls, for infinite loops, for variables used but not defined (and vice versa), and for errors in defining integer variables as real (and vice versa). Static analysis was performed once all of the simulation code was written, and it uncovered some errors that would have been difficult to find in any other way.

The second step was "instrumentation analysis" of major portions of the simulation code. Instrumentation analysis places probes in each logical branch of the code, and after the simulation has been run through a number of different test cases, it provides a record of the branches that have and have not been executed. In the TASRAN modules subjected to instrumentation analysis, about 75 percent of the branches were executed. Each branch not executed in the test cases was examined to determine why, and in virtually every case the branch was an error path. That is, only if the TASRAN data were in error would those paths be taken.

The final stage of testing was demonstration testing. A number of test cases were defined which exercise the major features of the simulation—using different track filters, track initiation algorithms, etc. Each of these cases was run to turn up any "bugs" in the simulation. Naturally some bugs turned up, typically as the simulated sites became saturated with more targets than their defined capacity. All of the bugs uncovered thus far have been fixed, so that the demonstration cases all run without any apparent problems.

Since TASRAN provides the option to run in an interactive mode with the user providing inputs as the simulation proceeds, extra efforts

were made to make the interactive portions of TASRAN "idiot proof."
That is, all of the interactive input requests define the options avail-
able for inputs, and the simulation checks the inputs from the user to
make sure that they are reasonable. Input errors cause TASRAN to ask
again for the input; in no cases tested to date have erroneous TASRAN
interactive inputs caused the program to fail, although a really crea-
tive user could probably find a way.

A version of TASRAN identical to that provided to RADC is main-
tained on GRC's computer. Many test cases have consistently provided
simulation results identical between the two computers to the extent
possible with machines of different word length. In particular, the
random number generators produce identical sequences. Consequently, any
errors that arise in TASRAN runs at RADC should be possible to duplicate
and trace down at GRC.

1.4.3 Possible Upgrades

Although TASRAN was developed to provide a basic capability for
evaluating netting concepts, a number of upgrades would make it even
more useful as an analysis tool. Some of these include:

- Airborne surveillance radars, such as AWACS or an airborne
  phased array.

- Algorithms for self-survey within the net, using common
  tracking of targets among several sites to determine rela-
  tive locations of the radars.

- Incorporation of models for target identification using data
  from other types of sensors along with the radar.

- Simplification of the radar and data processing modules to
  reduce the detail of the models for a single site while
  retaining the detailed models for interaction among multiple
  sites. This could reduce simulation running time.

2   TASRAN MODELING

This section provides details about some of the new models developed for the scenario, radar, and data processing modules of TASRAN. Since most of the communication module was borrowed intact from the Tactical Communication System model developed by GRC for HQ USAF/SA, details of the many communication models are described elsewhere.[1]

2.1   SCENARIO MODELS

2.1.1 Aircraft Flight Models

TASRAN uses different flight models for generally straight-and-level flight and for maneuvering flight. Both models generate a table of position and velocity vectors at fixed time steps along the flight path. Then aircraft position and velocity vectors at any given time are found by quadratic interpolation within the table. That is, suppose the ith row of the table gives the position vector $\bar{x}_i$ and velocity vector $\bar{v}_i$ at time $i\Delta$. Then the flight model returns the position vector for time t as

$$\bar{x} = \bar{x}_{i-1} + u(\bar{x}_i - \bar{x}_{i-1}) + \frac{u(u-1)}{2} (\bar{x}_{i-1} + \bar{x}_{i+1} - 2\bar{x}_i)$$

where $i\Delta$ is the time closest to t and

$$u = \frac{t - (i-1)\Delta}{\Delta}$$

The velocity vector $\bar{v}$ is found by the same formula, substituting $\bar{v}_i$ for $\bar{x}_i$.

---

[1] Dennis Cooper, Alan Cordes, and George Thompson, Command Enhancement Module (CEM) User's Manual, General Research Corporation CR-3-944, April 1981.

12

The tables are updated periodically so that at least half of the table entries are valid for future times.

Two separate flight models are used in generating the flight tables. The great-circle (GCIR) model is used for aircraft flight along a succession of great-circle segments. The aircraft speed is fixed along each segment, and the altitude is either constant or changing linearly along the segment. At the end of one segment, the aircraft abruptly turns to fly along the next segment. The abrupt turn is smoothed out by the quadratic interpolation in the table.

The great-circle flight model uses the concept of rotating position and velocity vectors in the plane defined by the two endpoint vectors. Suppose the great-circle segment is defined by two vectors $\overline{P}_1$ and $\overline{P}_2$ as shown in Fig. 2.1. In general $\overline{P}_1$ and $\overline{P}_2$ correspond to different aircraft altitudes h_1 and h_2. For this segment a special earth-center coordinate system is defined so that x' lies along $\overline{P}_1$, z' lies along the direction of $\overline{P}_1 \times \overline{P}_2$, and y' forms a right-hand system.

Vectors $\overline{P}_1$ and $\overline{P}_2$ form an angle θ_{12}. Thus the approximate length of the great-circle path is

$$\ell = \theta_{12}[R_e + (h_1 + h_2)/2]$$

where R_e is the earth's radius. If the airplane flies at speed v, the total time to traverse the segment is ℓ/v. At any time t, the angle θ_t between the aircraft position vector $\overline{P}$ and the initial vector $\overline{P}_1$ is given by

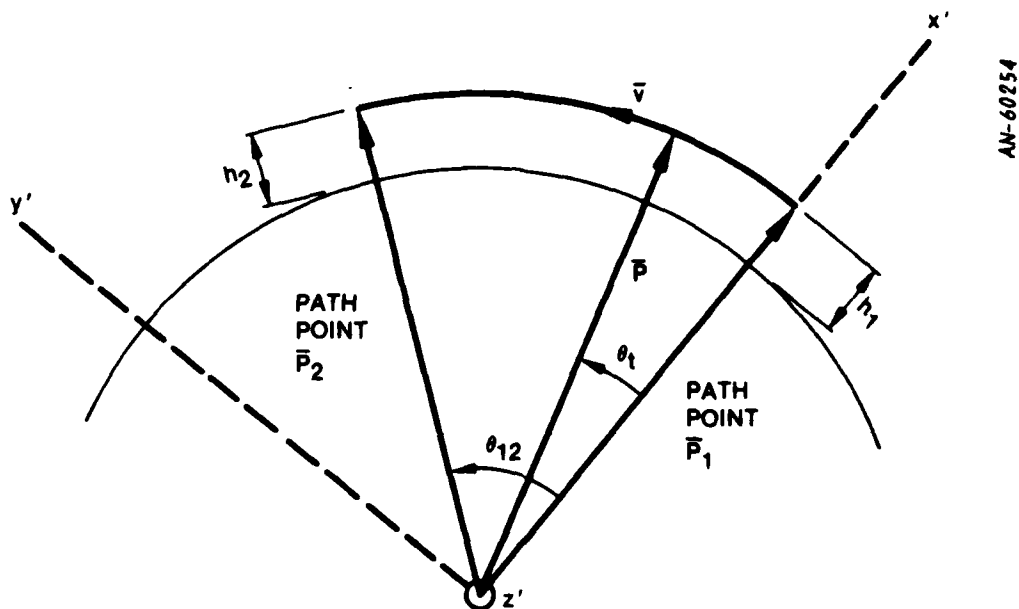$$\theta_t = \frac{vt}{[R_e + (h_1 + h_2)/2]}$$

13

AN-60254

Figure 2.1. Coordinates for a Great-Circle Flight Segment

Then in this special earth-center system the components of the position and velocity vectors are given by

$$P_{x'} = P \cos \theta_t$$

$$P_{y'} = P \sin \theta_t$$

$$P_{z'} = 0$$

$$v_{x'} = v \sin \theta_t$$

$$v_{y'} = v \cos \theta_t$$

$$v_{z'} = 0$$

14

where

$$P = R_e + h_1 + (h_2 - h_1)\theta_t/\theta_{12}$$

Finally the vectors $\overline{P}$ and $\overline{v}$ are transformed into the basic earth-center coordinate system used in TASRAN.

The pursuit-navigation (PNAV) model is used for aircraft maneuvering through a succession of closely spaced points, as shown in Fig. 2.2. This model commands turn rates proportional to the errors in heading angles to the next point, and axial accelerations to null out errors between actual and desired speeds. However, the acceleration is limited to a maximum value defined for the aircraft, and if the drag generated

- FLIGHT DESCRIBED BY INITIAL STATE AND A FEW LATER POINTS (AND SPEEDS)
- AIRCRAFT DESCRIBED BY MASS, AREA, DRAG, THRUST AND MAXIMUM G'S
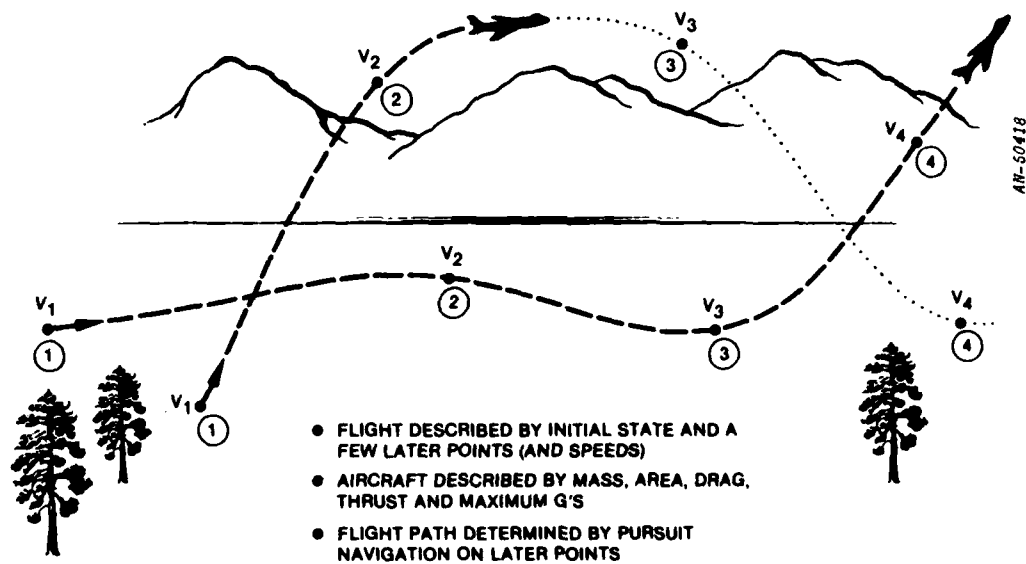- FLIGHT PATH DETERMINED BY PURSUIT NAVIGATION ON LATER POINTS

Figure 2.2. Pursuit Navigation Flight Model

by the aircraft turn exceeds the aircraft's maximum thrust, the airplane slows down.

The PNAV flight model uses a local Cartesian east, north, up coordinate system to compute pitch and yaw rates and atmospheric drag. However, all entries in the flight table are maintained in an earth-center Cartesian coordinate system.

The equations of motion are defined in the following way: first consider the aircraft at the center of a local east, north, up Cartesian coordinate system. Let the position of the destination point be defined by vector $\overline{R} = (R_x, R_y, R_z)$, the current velocity by vector $\overline{v} = (\dot{x}, \dot{y}, \dot{z})$, and the desired speed at the destination by scalar s. (Refer to Fig. 2.3.) If we define the yaw angle ψ measured from the x-axis and the pitch angle θ measured from the x-y plane, the errors in yaw and pitch between the current velocity vector $\overline{v}$ and the destination vector $\overline{R}$ are given by

$$\varepsilon_\psi = \tan^{-1}\left(\frac{R_y \dot{x} - R_x \dot{y}}{R_x \dot{x} + R_y \dot{y}}\right)$$

$$\varepsilon_\theta = \tan^{-1}\left(\frac{R_z \sqrt{\dot{x}^2 + \dot{y}^2} - \sqrt{R_x^2 + R_y^2}\ \dot{z}}{\sqrt{\dot{x}^2 + \dot{y}^2}\ \sqrt{R_x^2 + R_y^2 + R_z \dot{z}}}\right) + R/R_e$$

where R = magnitude of vector $\overline{R}$

R_e = earth's radius

The R/R_e term compensates for the earth's curvature, since in level flight the destination appears lower in pitch angle than it actually is. Pitch and yaw turn rates are then commanded to null out the errors as rapidly as possible

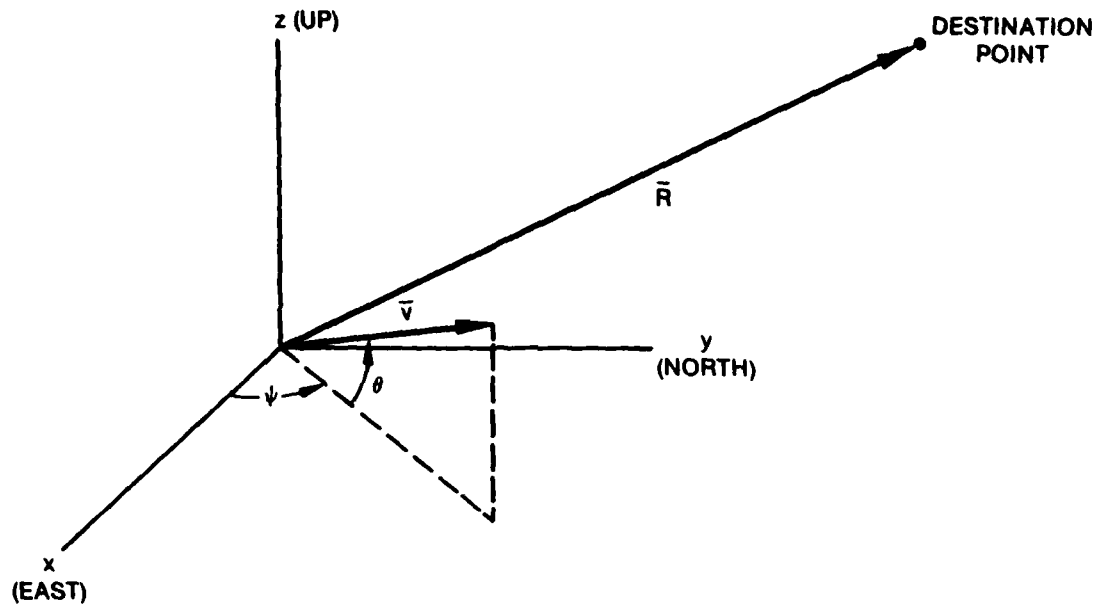Figure 2.3.   Local ENU Coordinate System for Pursuit Navigation Flight
Model

$$\dot{\psi}_c = \eta \epsilon_\psi$$

$$\dot{\theta}_c = 0.25 \eta \epsilon_\theta$$

where $\dot{\psi}_c$ = commanded yaw rate, rad/s

$\dot{\theta}_c$ = commanded pitch rate, rad/s

$\eta = a_{max}/v$ = navigation gain

a_{max} = maximum maneuver acceleration, m/s^2

v = current speed = magnitude of vector $\bar{v}$, m/s

17

For these commanded turn rates, the actual components of normal acceleration would be

$$a_\psi = v\dot{\psi}_c / \cos \theta$$

$$a_\theta = v\dot{\theta}_c - g \cos \theta$$

If $a_n = \sqrt{a_\psi^2 + a_\theta^2}$ exceeds a_{max}, the commanded pitch and yaw rates are scaled down so that $a_n = a_{max}$.

Finally the axial acceleration is computed. The desired velocity rate $\dot{v}_c$ is given by

$$\dot{v}_c = \frac{v}{R} (s - v)$$

However, the airplane may not be able to generate enough thrust to meet this desired acceleration. The drag and gravitational axial acceleration is given by

$$a_a = -g \sin \theta - \frac{1}{2} \rho v^2 A C_{DO} / m$$

$$- b m a_n^2 / \left(\frac{1}{2} \rho v^2 A \right)$$

where

g = gravitational acceleration, m/s^2

ρ = air density, kg/m^3

A = reference area of aircraft, m^2

C_{DO} = zero-lift drag coefficient

18

$$m = \text{aircraft mass, kg}$$

$$b = \text{induced drag coefficient}$$

$$a_n = \text{normal acceleration, m/s}^2$$

$$= \sqrt{a_\psi^2 + a_\theta^2}$$

Finally, $\dot{v}_c$ is adjusted if necessary to satisfy the following inequalities

$$\dot{v}_c > a_a$$

$$\dot{v}_c < T_{max}/m$$

$$\dot{v}_c > -3g$$

where T_{max} = maximum aircraft thrust, N.

Finally given $\dot{\psi}_c$, $\dot{\theta}_c$, and $\dot{v}_c$, the new position and velocity vectors and the pitch and yaw angles at the end of an integration step of τ seconds are given by

$$\bar{x}' = \tau \bar{v}$$

$$\theta' = \theta + \tau \dot{\theta}_c$$

$$\psi' = \psi + \tau \dot{\psi}_c$$

$$v' = v + \tau \dot{v}_c$$

$$\dot{x}' = v' \cos \theta' \cos \psi'$$

$$\dot{y}' = v' \cos \theta' \cos \psi'$$

$$\dot{z}' = v' \sin \theta'$$

$$\bar{v}' = (\dot{x}', \dot{y}', \dot{z}')$$

This integration proceeds in steps not greater than 0.25 second. The orientation of the local east, north, up coordinate system is recomputed for each entry in the flight table, say about once each 5 seconds of flight.

2.1.2 RCS Model

The radar cross section model provides for variation of average RCS with aspect angle. The average RCS is given

$$\bar{\sigma} = \sigma_{nose} \cos^2 \theta + \sigma_{side} \sin^2 \theta$$

where θ is the aspect angle off the nose of the aircraft.

A Swerling I fluctuating radar cross section is modeled as

$$\bar{\sigma} = \alpha \bar{\sigma} ,$$

where α is a random number drawn from an exponential distribution $P(\alpha) = 1 - \exp(-\alpha)$.

2.2 RADAR MODELS

2.2.1 Target Returns

The positions of targets maintained by the Scenario Module are updated to the time of each radar measurement. Then for each target in the antenna beam (search mode) or in the range gate (track mode), the target range, azimuth and elevation angles, and range-rate are computed. These values are then used to compute the target's received signal power

20

$$S = \frac{PG^2\lambda^2\sigma}{(4\pi)^3 R^4 L}$$

where

       P = peak transmit power, W

       G = radar antenna gain

       λ = radar wavelength, m

       σ = target radar cross section, m^2

       R = target range, m

       L = system losses

The target return signal power competes with three sources of interference—thermal noise that is always present, noise jamming that may be present, and clutter. The interference for each target is combined into a single interference value

$$I = N + J + C$$

where

       N = random thermal noise power drawn from exponential distribution with mean kTB (see next section)

$$J = \frac{P_j G_j G\lambda^2 B}{(4\pi)R^2 B_j^2} = \text{jammer noise power}$$

$$C = \frac{\alpha P_t G_c^2 \lambda^2 \sigma_c R\Delta\theta\Delta R}{(4\pi)^3 R^4 L} = \text{clutter power}$$

and where k = Boltzmann's constant

       T = radar noise temperature, K

       B = radar receive bandwidth, Hz

P_j = jammer average power, W

G_j = jammer antenna gain

B_j = jamming bandwidth, Hz

α = clutter rejection factor (MTI, etc.)

G_c = radar antenna gain in direction of clutter patch

σ_c = clutter reflection coefficient, m^2/m^2

$\Delta\theta$ = radar azimuth resolution

ΔR = radar range resolution

Then S/I is compared to a threshold to determine target detectability and to determine the accuracy of the measurement as discussed subsequently.

These factors are either constants provided as simulation inputs or values calculated using antenna pattern models and the radar-target geometry. The clutter rejection factor α , however, depends on target range rate and the clutter rejection capability of the radar. The clutter rejection factor is modeled in the following simple way:

$$\alpha = 1 \quad , \quad 0 \le \dot{R} < V_1$$

$$= \alpha_o \quad , \quad V_1 \le \dot{R}$$

Thus no clutter rejection is obtained for a target with too low range rate, while rejection of $-10 \log_{10}(\alpha_o)$ decibels is achieved for targets with higher velocities. The constants V_1 and α_o are inputs for each radar.

2.2.2 Noise Model

The noise model provides three functions: it degrades measurement accuracy in accordance with the signal-to-interference ratio; it

22

influences the probability of target detection; and it creates false alarms.

Simulated measurements are constructed from exact values of the measurements, and then corrupted by additive random error terms. For example, if the exact value of range is R , then the simulated value is given by $R + n$, where n is a zero-mean, normal random variable generated by a random number subroutine. The variance of n is determined by $\sigma_n^2 = (\Delta R)^2 [k_1 + k_2/(S/I)]$, where ΔR is the range resolution, k_1 is a bias term, and k_2 is a "range-splitting" term. Typical values for k_1 and k_2 are $(0.05)^2$ and $(0.1)^2$, respectively. For other measured values such as azimuth and elevation, their resolution and appropriate values of k_1 and k_2 are substituted.

Thermal noise power is simulated by a random number N drawn from an exponential distribution with mean kTB . (Noise voltages that have a Gaussian distribution produce an exponentially distributed power.) This random noise power N is added to the jamming power J and clutter power C in computing the total interference I.

False alarms occur randomly but at an average rate in the radar. Times for false alarms are drawn from a Poisson distribution matching the false alarm rate. Then range and elevation for the false alarm detection are drawn from uniform distributions with the limits $0 < R < R_{max}$ for range and $-\frac{\Delta\theta}{2} \le \theta \le \frac{\Delta\theta}{2}$ for elevation. The azimuth is given by the radar scan angle at the false alarm time.

2.2.3 Antenna Pattern

The antenna pattern model used for both reflector and phased-array antennas is illustrated in Fig. 2.4. The parameters of the model are the peak gain, the beamwidth, the near-in sidelobe level and the angle at which it begins, and the backlobe level and the angle at which it begins. The mainlobe has a $\sin(x)/x$ shape, the sidelobes vary with

23
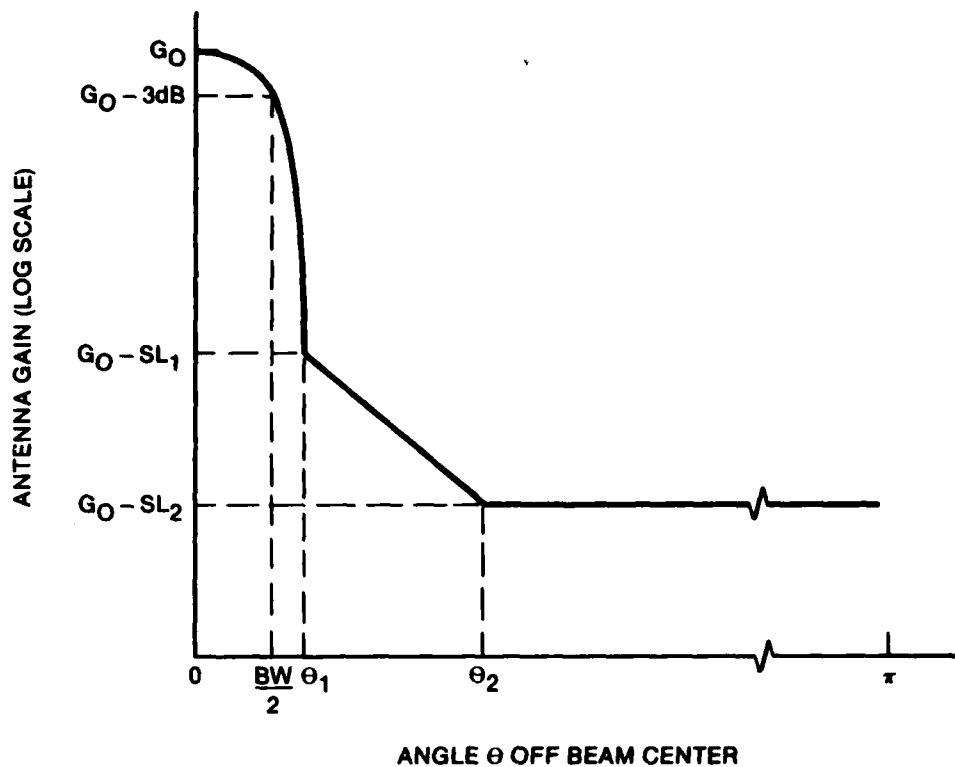
Figure 2.4. One-Dimensional Antenna Pattern Model

angle between the peak sidelobe level and the backlobe level, and the backlobe level is constant. The pattern model in the figure applies to either the elevation or azimuth plane in a reflector antenna or to either the u-plane or v-plane in an array. The two-dimensional pattern is given by a product

$$G(\theta,\phi) = G_1(\theta)G_2(\phi)$$

where G_1 and G_2 are modeled as shown in the figure.

2.3   DATA PROCESSING MODELS

2.3.1   Track Initiation

Track initiation is the process in which a sequence of returns from newly observed targets are stored and associated (correlated) to

24

establish a track. All new radar returns—those that do not associate
with an existing track—are candidates for track initiation. Some of
these returns are false alarms due to noise and clutter. One of the
functions of the track-initiation process is to eliminate these false
alarms before tracks are established on them.

The track-initiation process involves the association of measure-
ments of target position and possibly velocity to determine whether or
not the returns are from the same target and if so to obtain an estimate
of the target position and velocity for track-filter initialization.
Track initiation can be carried out using different types of measure-
ments, numbers of measurements, and measurement spacings, with different
levels of performance associated with each of the variations. TASRAN
provides two alternative track initiation procedures, based respectively
on two and three radar measurements.

Two Measurements. The simplest track-initiation algorithm asso-
ciates two measurements[1] of target position, made, for example, on suc-
cessive scans of a TWS radar. The algorithm takes each measurement that
does not associate with an existing track and first determines if it
associates with a previously saved measurement. The association test is
based on speed; if two measurements produce a track with a credible
speed (or velocity vector)—i.e., a speed that is not greater than the
maximum speed of any aircraft of interest at the target altitude—a
track is initiated with the second measurement as the estimated position
and a velocity based on the two position measurements and the time
between them. Tracks are initiated on all pairs of measurements that
pass the speed test. Whether or not a measurement is used to initiate a

---

[1]Each measurement may involve a number of radar pulses which are trans-
mitted and received during the time the target is within the antenna
beam. In the AN/TPS-43 radar, for example, about 8 pulses are trans-
mitted during the time the antenna beam moves one beamwidth. The
returns from these pulses are integrated by the radar signal processor
to produce a single measurement. Only the results of this integration
will be simulated.

track, it is saved for possible association with later measurements;
this ensures that measurements on valid targets are not lost during the
track initiation process by an accidental association with a false
alarm.

Three Measurements. Improved performance can be obtained by using
three measurements on successive scans or three more closely spaced
measurements in a computer-directed-track system. The first two meas-
urements are used to set up a tentative track. The algorithm takes each
measurement that does not associate with an existing track and first
determines if it associates with a tentative track. Association is
based not on speed, but on the same association tests used to determine
if a measurement associates with an existing track, as discussed subse-
quently. An association volume is placed around each extrapolated
track, and association is successful if the measurement lies inside the
volume. If association is successful, the measurement is used to
initiate a new track. If it is not successful, the procedure described
above under "two measurements" is followed.

2.3.2 Track Association

The process of track association (correlation) is central to the
operation of any system which must detect and track large numbers of
targets. Failure to associate a radar return with the proper target
track--or to determine correctly that it does not associate with any
track in the file--can cause duplicate tracks to be established, intro-
duce large errors in the track estimate, and possibly cause a loss of
track. The latter consequence leads at best to increased radar and data
processor loads for repeated track initiations, and at worst to the loss
of needed information about the target and its flight path.

Track association is similar to track initiation in that a current
measurement of target position is compared with the expected target
position based on previous measurements. The principal difference is

26

that in the association process the expected target position is predicted from a number of previous radar measurements which have been appropriately combined and filtered, rather than from only one or two previous track-initiation measurements.

The probability of correctly associating a radar return with the proper track depends on the magnitudes of the measurement error, the track prediction error, and the deviation of the target position from the predicted track due to maneuvers initiated since the last measurement, as well as on the association algorithm used and on the density of false alarms and other targets in the vicinity of the target of interest.

The TASRAN association algorithm associates each radar return with the closest track within a specified association volume. More precisely, this algorithm consists of the following steps: (1) extrapolate each track of potential interest to the time of the radar measurement; (2) determine the distance from the measured target position to each of the predicted target positions which lie within a previously specified association volume (the distance could be in two, three, or four dimensional space depending on the dimensionality of the radar measurements); and (3) associate the radar return with one of the tracks based on the calculated distances. In situations where there are several tracks within the association-volume of the measurement, the track closest to the measurement is associated with that measurement.

A simple association scheme is illustrated in Fig. 2.5. A target position is extrapolated to the time of the measurement, and one or more circles (or spheres in three dimensions) are put around the extrapolated position as shown. If the measured point is outside the outer circle (as in P_3), the measurement does not associate. The radius of the outer circle is chosen so that it encompasses any possible maneuver plus errors due to noise.

CASES:

P$_1$ ASSOCIATES, NONMANEUVERING

P$_2$ ASSOCIATES, MANEUVERING

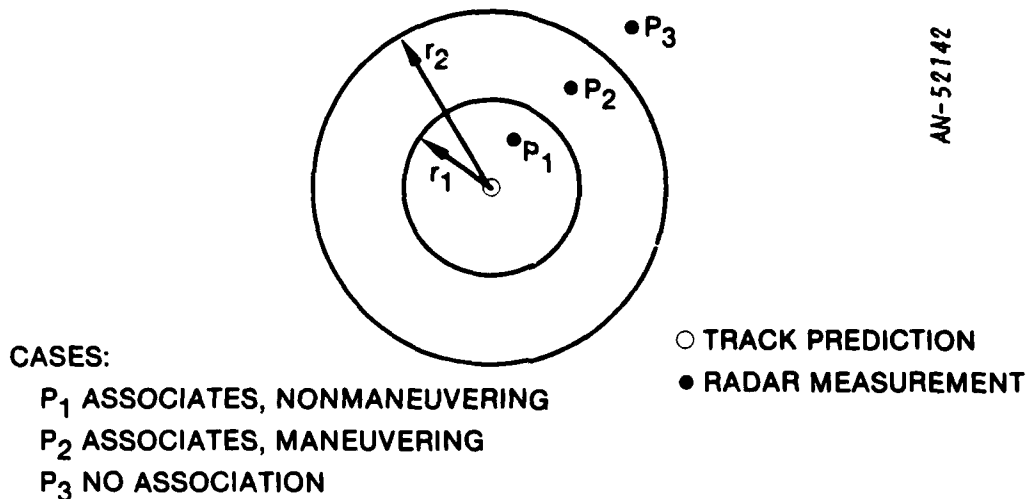P$_3$ NO ASSOCIATION

○ TRACK PREDICTION

● RADAR MEASUREMENT

Figure 2.5. Closest-Pair Actual-Distance Algorithm

In some tracking filters an estimate of whether or not the target is maneuvering is required. The inner circle is used for this purpose. If the measurement point is between r_1 and r_2 (as in P$_2$), the target is maneuvering. If the distance is less than r_1 (as in P$_1$), the target is not maneuvering. The threshold r_1 is chosen so that it encompasses only a very modest maneuver plus measurement noise.

2.3.3 Track Update (Filtering)

Although a number of different track filters are potentially applicable to the netted surveillance system, TASRAN provides two different filters: the α-β filter and the Kalman filter. The α-β filter is simpler than the Kalman filter, and requires less simulation time to execute. The Kalman filter, however, provides an estimate of the accuracy of the track, and inherently provides an optimum weighting of noisy measurement data. Both filters are recursive, so that previous track data need not be saved.

Kalman Filter. The Kalman filter is the optimal filter for linear systems perturbed by Gaussian noise for several reasonable optimization criteria such as minimum mean square error, maximum likelihood, and minimal variance Bayes' estimate. The extended Kalman filter has proven useful in many non-linear problems as well, and has been widely studied for use in tracking maneuvering aircraft.

The Kalman formulation offers some advantages over simpler algorithms. For example, it provides a state error covariance matrix which can be used to properly combine tracks from separate radars into a single System Track. This same covariance matrix can also be used for statistical association between tracks. Also provided is the measurement-residual covariance matrix for use in statistical association between measurements and tracks.

The disadvantages of the Kalman filter are (1) the large amount of data processing resources required to implement the algorithm, and (2) the increased communications bandwidth required to transmit the state error covariance matrix.

The Kalman filter in TASRAN is the 6-state (3-position, 3-velocity) filter using maneuver noise to model acceleration. The logical structure and generalized equations for the Kalman filter are summarized in Fig. 2.6.

α-β Filter. The α-β filter is a greatly simplified recursive filter which has been extensively investigated and widely used. In the α-β filter the state vector consists of position and velocity:

AN-50425

SYSTEM MODEL

MEASUREMENT MODEL

KALMAN FILTER
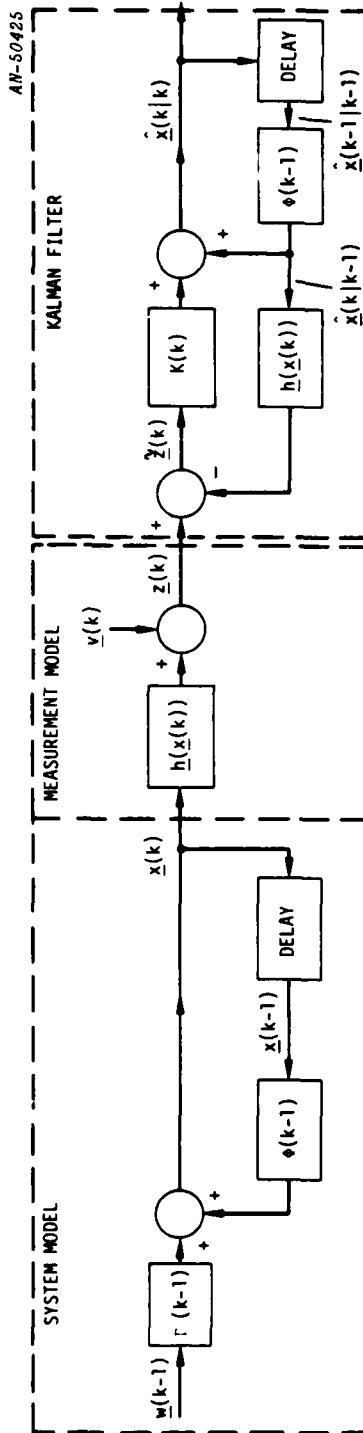
- SYSTEM MODEL

$$\underline{x}(k) = \phi(k-1)\underline{x}(k-1) + \Gamma(k-1)\underline{w}(k-1)$$

$\underline{x}(k)$ = SYSTEM STATE AT TIME k

$\phi(k)$ = STATE TRANSITION MATRIX

$\underline{w}(k)$ = MODEL (OR MANEUVER) NOISE $\sim N(\underline{0}, Q(k))$

$N(\underline{m}, Q)$ = NORMAL DISTRIBUTION WITH MEAN $\underline{m}$ AND COVARIANCE Q

- MEASUREMENT MODEL

$$\underline{z}(k) = \underline{h}(\underline{x}(k)) + \underline{v}(k)$$

$\underline{z}(k)$ = MEASUREMENT AT TIME k

$\underline{h}(\underline{x}(k))$ = SYSTEM STATE IN MEASUREMENT COORDINATES

$\underline{v}(k)$ = MEASUREMENT NOISE $\sim N(\underline{0}, M(k))$

- FILTER EQUATIONS

$$\underline{\hat{x}}(k|k) = \underline{\hat{x}}(k|k-1) + K(k)\underline{\tilde{z}}(k)$$

$\underline{\hat{x}}(k|k)$ = ESTIMATE OF $\underline{x}(k)$ GIVEN k MEASUREMENTS

$\underline{\hat{x}}(k|k-1)$ = ESTIMATE OF $\underline{x}(k)$ GIVEN $k-1$ MEASUREMENTS

$\underline{\tilde{z}}(k)$ = MEASUREMENT RESIDUAL

$K(k)$ = KALMAN GAIN MATRIX

$$\underline{\tilde{z}}(k) = \underline{z}(k) - \underline{h}(\underline{\hat{x}}(k|k-1))$$

$$\underline{\hat{x}}(k|k-1) = \phi(k-1)\underline{\hat{x}}(k-1|k-1)$$

Figure 2.6. Kalman Filtering (Discrete Form)

$$\hat{\underline{x}} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}$$

$$\hat{\dot{\underline{x}}} = \begin{bmatrix} \hat{\dot{x}} \\ \hat{\dot{y}} \\ \hat{\dot{z}} \end{bmatrix}$$

These are extrapolated from time t_{k-1} to time t_k by

$$\hat{\underline{x}}(k|k-1) = \hat{x}(k-1|k-1) + \hat{\dot{\underline{x}}}(k-1|k-1)T$$

$$\hat{\dot{\underline{x}}}(k|k-1) = \hat{\dot{\underline{x}}}(k-1|k-1)$$

where

$\hat{\underline{x}}(k|k-1)$ = extrapolated position vector at time t_k using
measurements through time t_{k-1}

$\hat{\dot{\underline{x}}}(k|k-1)$ = extrapolated velocity vector

$\hat{\underline{x}}(k-1|k-1)$ = estimated position at time t_{k-1}

$\hat{\dot{\underline{x}}}(k-1|k-1)$ = estimated velocity at time t_{k-1}

$$T = t_k - t_{k-1}$$

The position is updated by

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + \alpha \left| \underline{x}_m(k) - \hat{\underline{x}}(k|k-1) \right|$$

31

where $\underline{x}_m(k)$ is the measurement in Cartesian coordinates at time t_k and α is a constant between 0 and 1. Note that the quantity $[\underline{x}_m(k) - \hat{\underline{x}}(k|k-1)]$ is the measurement residual in Cartesian coordinates.

If $\alpha = 0$, the measurement is ignored and the new estimate is the extrapolated position; if $\alpha = 1$, the new estimate is the measurement, and the extrapolated position is ignored. Usually α is somewhere between, and some of both the extrapolated position and the measurement are used; this is illustrated in Fig. 2.7a.
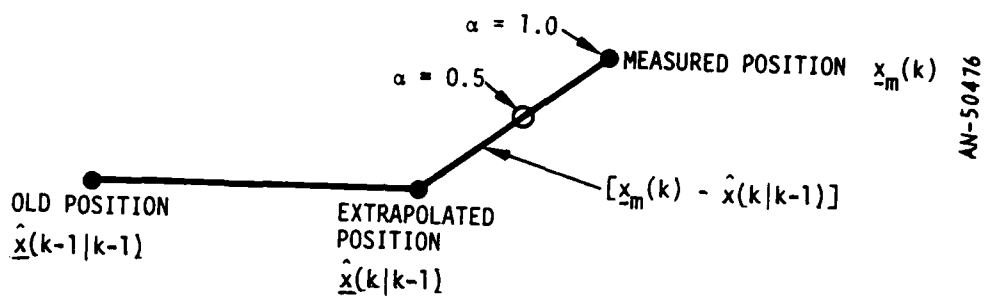
The velocity is updated by

$$\hat{\dot{\underline{x}}}(k|k) = \hat{\dot{\underline{x}}}(k|k-1) + \frac{\beta}{T}\left[\underline{x}_m - \hat{\underline{x}}(k|k-1)\right]$$
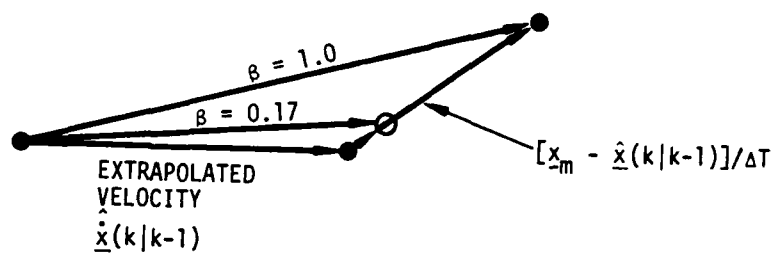
where β is a constant between 0 and 1. Note that $[\underline{x}_m - \hat{\underline{x}}(k|k-1)]/T$ is the velocity residual. The quantity β determines how much of this residual will be added to the extrapolated velocity. There is an optimum value of β in terms of α; thus there is only one free parameter to choose from. This relation is

$$\beta = \alpha^2/(2 - \alpha)$$

Maneuver detection is sometimes used with α-β filters. This can be based on the size of the measurement residual. If the aircraft is flying along a straight path, the past measurements that went into the extrapolated position are still valid, and a small value of α can be used. If a maneuver is detected, the present measurement should be weighted more heavily, and a value of α near 1.0 is used.

32

$\alpha = 1.0$

$\alpha = 0.5$

MEASURED POSITION $\underline{x}_m(k)$

$[\underline{x}_m(k) - \hat{x}(k|k-1)]$

OLD POSITION
$\hat{\underline{x}}(k-1|k-1)$

EXTRAPOLATED
POSITION
$\hat{\underline{x}}(k|k-1)$

- $\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1)] + \alpha[\underline{x}_m(k) - \hat{\underline{x}}(k|K-1)]$

$\beta = 1.0$

$\beta = 0.17$

$[\underline{x}_m - \hat{\underline{x}}(k|k-1)]/\Delta T$

EXTRAPOLATED
VELOCITY
$\dot{\hat{\underline{x}}}(k|k-1)$

- $\dot{\hat{\underline{x}}}(k|k) = \dot{\hat{\underline{x}}}(k|k-1) + \beta[\underline{x}_m - \hat{\underline{x}}(k|k-1)]/\Delta T$

NON-MANEUVERING FILTER

$\alpha = 0.5$ $\qquad \beta = \dfrac{\alpha^2}{2 - \alpha} = 0.17$

MANEUVERING FILTER

$\alpha = 1.0$ $\qquad \beta = 1.0$

Figure 2.7. $\alpha-\beta$ Filter Illustration

33

2.3.4 Track File Maintenance

Track file maintenance has two basic functions: to purge old tracks and to combine duplicate tracks. Old tracks may be purged if no update has occurred on them for a specified number of seconds. A periodic pass through the Local Track File, testing each Local Track dataset for its most recent update time, should handle the purge operation. Old datasets are merely removed from the file and destroyed.

Combining duplicate tracks is best handled in a TWS system by noting when a new radar return associates with more than one target. Then the two (or more) targets can be compared on either an absolute or statistical test of the difference in position and velocity. If the two appear to be the same, then one of the Local Track data sets is removed from the Local Track file and destroyed. The merging of the two targets into one necessitates a change in the System Track File and notification of all other net members which would also have duplicate tracks.

2.3.5 Radar Scheduler

Each simulated phased-array radar is computer-controlled pulse by pulse. The radar is assumed to operate at a fixed pulse repetition frequency (PRF). The radar scheduler is invoked periodically to schedule pulses for the next time interval. The scheduler maintains time-ordered lists of requested track initiation and track measurements. At the time of each pulse, priority is given first to track initiation and then to track. If no track initiation or track request is valid at the time of the pulse, a search pulse at the next beam position in the surveillance raster is scheduled.

2.3.6 Surveillance Raster

The surveillance raster for each phased-array radar is defined by TASRAN inputs. The surveillance raster consists of a variable number of constant-elevation sweeps between specified azimuth limits. Each sweep

is defined by its elevation angle (for the center of the beam), the azimuth limits, the azimuth step size, and the minimum and maximum range limits. All of these parameters may be varied from sweep to sweep. The raster generator computes the azimuth and elevation of the next beam position, converts these angles into array u-v coordinates, and returns the beam position and range limits to the radar scheduler. The elevation sweeps are covered in a fixed order, and then the scan is repeated.

3   TASRAN TESTING

3.1   DEVELOPMENT TESTING

As the modules of TASRAN were developed, they were tested with special test programs or with debugging output statements embedded in the code. For example, the great circle and pursuit navigation flight models were tested by printing out the position and velocity of the aircraft at intervals of a few seconds along their flight paths. These values were compared to hand calculations of the correct aircraft position and velocity at those times. The radar models, the track initiation and track filter models, and the communication models were tested in the same way. These development tests provided confidence that the models were working properly.

Many of the debugging statements used in development testing remain in the TASRAN code. These may be turned on during the data initialization step if a detailed look at the operation of any particular model is desired in a simulation run.

3.2   RXVP80 TESTING

TASRAN was tested using RXVP80, a proprietary software tool developed in The Software Workshop at General Research Corporation. RXVP80 is a tool for analyzing source programs written in FORTRAN, IFTRAN (a structured version of FORTRAN developed by GRC), and IFTRAN dialects such as CIFTRAN, the language used for virtually all of TASRAN. RXVP80 is a successor to FAVS, the FORTRAN Automatic Verification System, used at RADC.

Two types of RXVP80 testing were done on TASRAN. The first was "static analysis," done when all of the TASRAN source code was first completed. Static analysis is used to detect coding inconsistencies by:

- Mode and type checking, which identifies misuses of constants and variables in expressions and assignments.

- Call checking, which compares function and subroutine calls with the formal parameter declarations, checking for consistency in the number of parameters, their mode (real or integer), and their type (single variable or dimensioned array).

- Set/use checking, which uncovers variables used before they are set, or set but never used. This frequently finds misspelled variables.

- Graph checking, which identifies possible problems in module structure such as unreachable code and infinite loops.

Static analysis revealed a few set/use errors, most of which were misspelled variable names. These set/use errors would have produced errors in TASRAN execution that would have been difficult to find with more conventional testing methods. All of these have been corrected. It also flagged a number of possible mode errors. Most of these occurred in handling character data in which the mode (real or integer) doesn't really matter as long as different modes are not mixed. All of the flagged errors were checked, and those that really could cause a problem were corrected.

Static analysis did not reveal any graph errors. However, the CIFTRAN translator would ordinarily trap errors of this type prior to RXVP80 testing.

Many of the more complex subroutines in TASRAN were subjected to RXVP80 instrumentation analysis. RXVP80 inserts into each program module a probe that counts the number of times that the module is executed. It also inserts into each logical branch of the module a probe that counts the number of executions of the branch. For example, it would put probes into both the "true" and "false" branches of a FORTRAN IF test. A special RXVP80 listing provides identifying numbers to each logical branch of the module.

37

After the TASRAN code was instrumented, a number of test cases designed to exercise most of the branches of the program modules were run. These test cases built a cumulative history of the execution of all of the instrumented branches of the program. This history file was then analyzed by RXVP80 to summarize the number of executions of each module and of each branch in the module, and to flag any logical paths not executed. By going back to the listings of the instrumented code, we examined each logical branch that was not executed. In a few cases we designed a new test case to exercise that branch; however, in many cases we found that the branch represents an error condition that doesn't require further testing.

A summary of the RXVP80 instrumentation testing of TASRAN is provided in Table 3.1. Of the 28 modules subjected to instrumentation testing, the total coverage of the logical branches in the test cases amounted to 877 out of 1148, or 76.4 percent. However, when the 53 branches that are clearly error conditions are excluded, the coverage amounts to 877 out of 1095, or 80.1 percent.

3.3   DEMONSTRATION TESTING

A number of netting system configurations and scenarios were developed to demonstrate the capabilities of TASRAN for demonstration/ acceptance testing at RADC.

The demonstration test cases are all based on the system configuration illustrated in Fig. 3.1. This configuration consists of two phased-array (or CDT) radars, seven conventional rotating surveillance radars operating in a track-while-scan (TWS) mode, and three operations control (or system) sites. Each system site is collocated with a TWS radar, and two of the system sites report to the third.

Figure 3.1 illustrates the defined regions of responsibility for each of the radars. The sectors for the CDT radars are their actual

TABLE 3.1

RXVP80 INSTRUMENTATION TESTING OF TASRAN

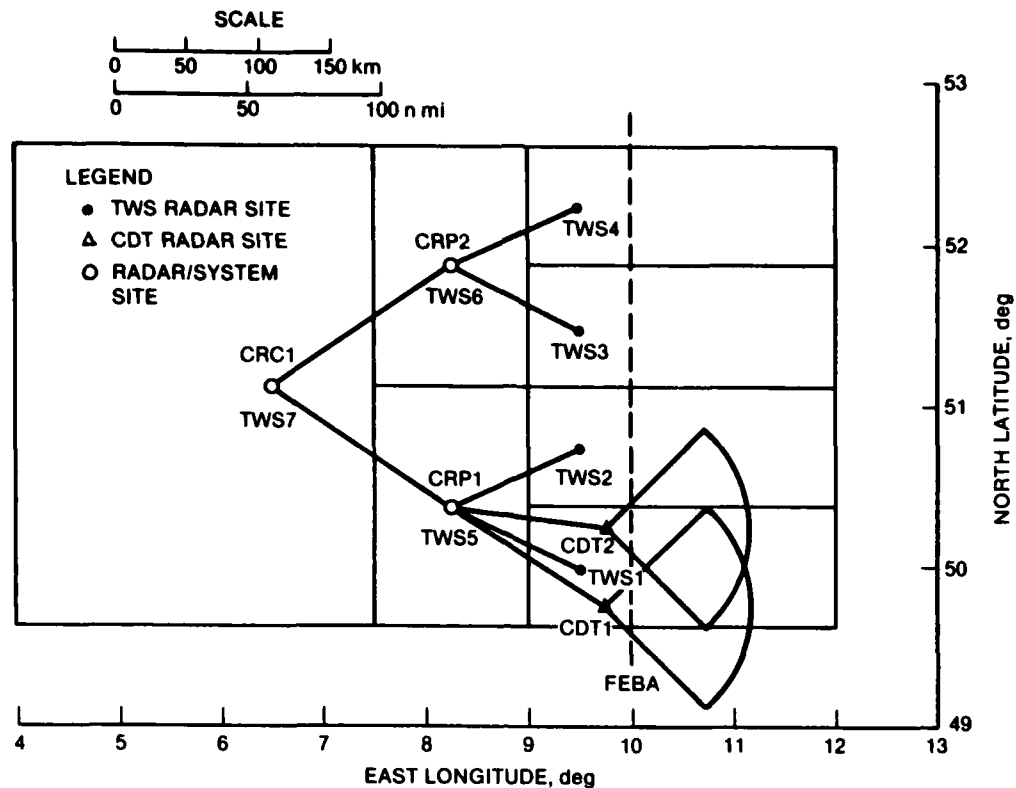| Subroutine	Number of Logical Branches	Number of Branches Tested	Percentage Tested	Number of Error Paths	Percentage Tested Excluding Error Paths
CDTANT	13	5	38.5	6	71.4
CDTASS	19	18	94.7	1	100.0
CDTINI	94	66	70.2	4	73.3
CDTMGT	34	31	91.2	0	91.2
CDTRAD	64	36	56.3	0	56.3
CDTRAS	5	5	100.0	0	100.0
CDTRET	47	35	74.5	4	81.4
CDTSKD	19	19	100.0	0	100.0
CDTTRK	58	53	91.4	3	96.4
EVPROC	52	37	71.2	4	77.1
FLYOBJ	80	57	71.2	5	76.0
OBJINT	5	4	80.0	0	80.0
OBJUPD	19	11	57.9	0	57.9
SETUP	118	79	66.9	12	74.5
SNDMSG	26	6	23.1	3	26.1
SYSASS	18	17	94.4	1	100.0
SYSMGT	56	51	91.1	0	91.1
SYSRET	56	51	91.1	2	94.4
SYSTRK	59	53	89.8	4	96.4
TWSASS	19	18	94.7	1	100.0
TWSDUP	34	18	52.9	0	52.9
TWSINI	74	61	82.4	1	83.6
TWSMGT	24	22	91.7	0	91.7
TWSRAD	62	45	72.6	0	72.6
TWSRET	12	12	100.0	0	100.0
TWSTRK	57	50	87.7	2	90.9
WHERE	18	15	83.3	0	83.3
WHEREJ	6	2	33.3	0	33.3
Total	1,148	877	76.4	53	80.1

Figure 3.1. Radar and System Sites for TASRAN Testing

coverage boundaries; targets outside of these sectors are not seen. However, the rectangular regions around the TWS radars define areas within which the radar tracks targets that it sees; it may see targets outside of the rectangle, but it ignores them.

Figures 3.2 and 3.3 illustrate the ground tracks of the aircraft included in the demonstration test cases. Figure 3.2 illustrates the ground track of a MiG-21 aircraft flying a maneuvering trajectory with the pursuit-navigation (PNAV) flight model. It also illustrates the ground track of a helicopter with two hover and vertical flight segments. A single fixed ground target is included in this figure as well. Figure 3.3 illustrates the ground track of all of the aircraft using the great-circle (GCIR) flight model. This includes two jamming aircraft flying parallel to the FEBA, plus three other aircraft, plus a mass raid
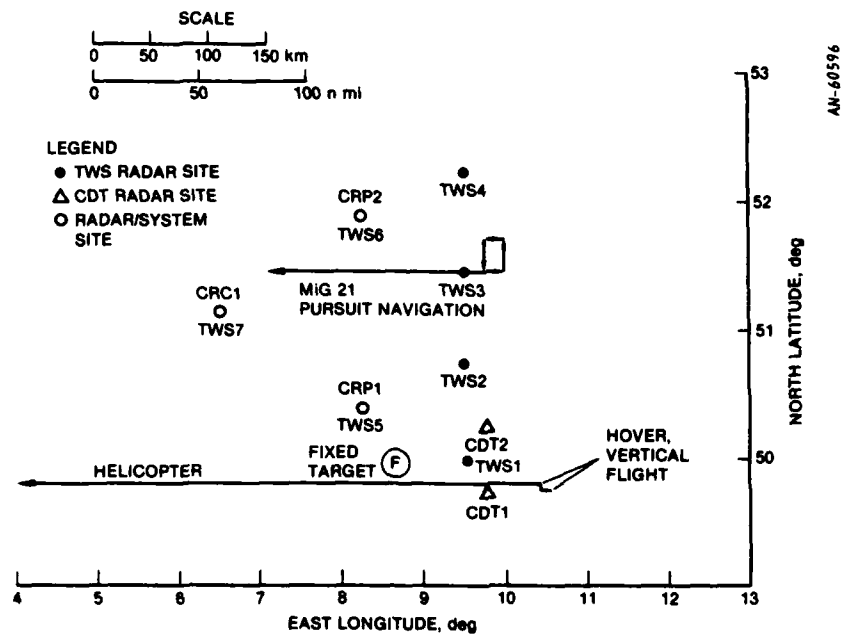
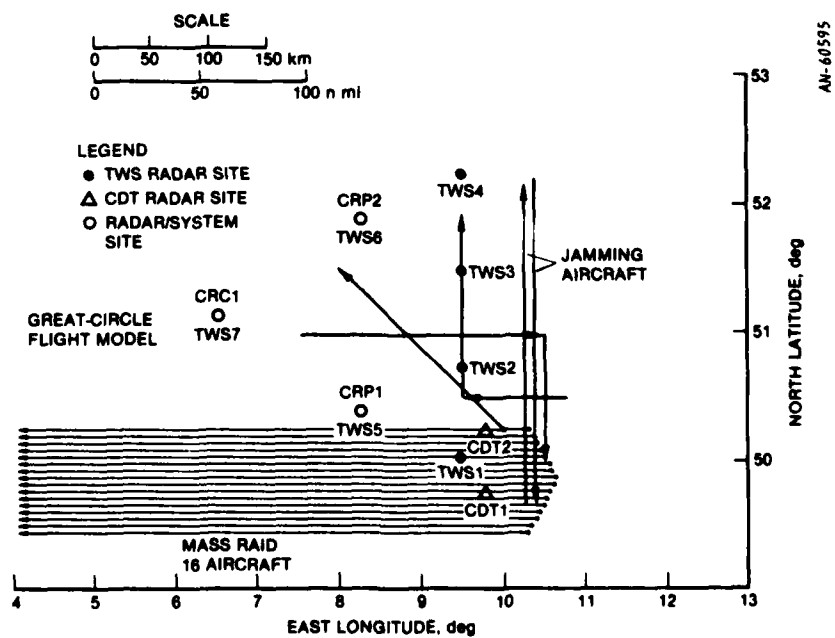**Figure 3.2. Helicopter and Pursuit Navigation Flight Paths**

Figure 3.3. Great-Circle Flight Paths

of sixteen aircraft flying parallel tracks at the same altitude. The
mass raid was designed to saturate some of the radar sites.

The communication network for the test cases is shown in Fig. 3.4.
The two CDT radars and the system site CRP1 are tied together in a JTIDS
broadcast network. All of the sites except the CDT radars are members of
a point-to-point troposcatter network. The relay node between sites CRP1
and CRC1 demonstrates the capability to relay messages, while sites CRP2
and CRC1 are tied together by both a tropo link and a backup HF radio
link. The Communication Module can work in three levels of com-
plexity—"NONE," which bypasses the Communication Module and passes mes-
sages instantaneously from site to site, "FAST," which models queuing and
transmission delays, and "TECH," which adds to the FAST model propagation
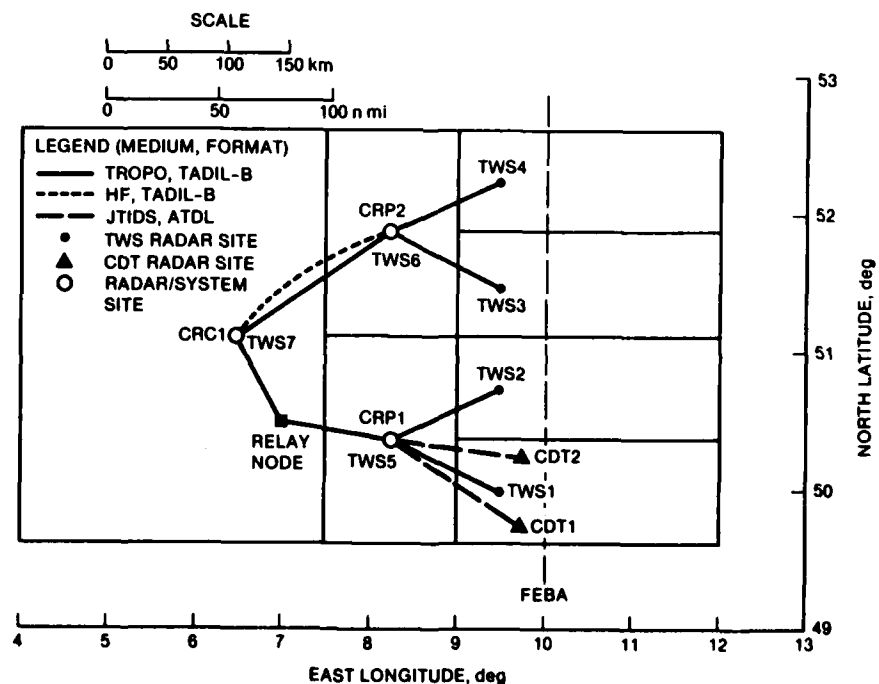and jamming degradation of message transmission.

Figure 3.4. Communication Network for TASRAN Testing

42

Table 3.2 indicates the various options within the six demonstration test cases. These cases exercise the options for track initiation, track filter, system track, communication mode, and jamming. All of the cases except Case 4 exclude the northernmost ten aircraft of the mass raid; all of the cases include the two jamming aircraft, but only in Case 3 are the jammers—one radar jammer and one communication jammer—turned

TABLE 3.2

TASRAN DEMONSTRATION CASES

	Case Number					
	1	2	3	4	5	6
Number of:						
Airplanes	12	12	12	22	12	12
Helicopters	1	1	1	1	1	1
Fixed Targets	1	1	1	1	1	1
Jammers	0	0	2	0	0	0
CDT Radar						
Track Init.	3 Hit	2 Hit	3 Hit	3 Hit	3 Hit	--
Track Filter	Kalman	α-β	Kalman	Kalman	Kalman	--
TWS Radar						
Track Init.	2 Hit	3 Hit	2 Hit	2 Hit	3 Hit	2 Hit
Track Filter	α-β	Kalman	α-β	α-β	Kalman	α-β
Boundaries	Yes	Yes	Yes	No	Yes	Yes
System Sites						
Track Filter	Select	Select	Select	Select	Merged	Select
Communication Mode	None	None	None	None	None	None
	Fast					Fast
	Tech					

43

on. Case 4 also eliminates the assigned boundaries about the TWS radars; this permits each radar to track more targets and causes the TWS sites near the FEBA to become saturated.

Case 6 omits the two CDT radars and provides an interesting case for demonstrating interactive displays without requiring long running times. The CDT radars, since they are controlled pulse by pulse, take a comparatively long time to run.

Case 1 has been used to demonstrate all three Communication Module modes; Case 6 has demonstrated the NONE and FAST options. However, any test case can be run in any mode by setting the proper flags during the database initialization step.

All of these test cases were run on GRC's computer system prior to TASRAN installation at RADC, and they revealed some problems that earlier test cases had not encountered.

Most of the problems were related to what happens when a simulated radar data processor becomes saturated. The maximum number of tracks at a site is set as an input parameter to TASRAN to model existing netting system limitations. When this limit is reached, and then new targets are encountered, the site data processing algorithms have to either drop old targets or ignore the new ones. Since earlier test cases had never included enough targets to saturate any site, the algorithms had never been tested, and, naturally, they failed. All of these errors have been corrected.

Two additional test cases were developed to demonstrate the handling of actual radar track messages. In Case 7, which used the scenario of Case 6, the Crosstell Simulation Module wrote out all messages from one radar/ data processor site (the TWS1 site) in a binary code onto a disk file. In Case 8, that radar/data processor site was omitted, and

its function was taken over by the Active Crosstell Module. The Active
Crosstell Module read the messages off the disk file, decoded them,
generated messages in the standard TASRAN message format, and passed them
along to the Communication Module. These messages were then practically
identical to the messages originally sent by the radar/data processor
site in Case 7. The only differences in the Case 7 and Case 8 messages
were due to truncation of precision in the coding process. The two test
cases produced essentially identical system tracks despite the two
different ways in which the track messages from the TWS1 site were
generated.

4   POSSIBLE UPGRADES TO TASRAN

Since TASRAN was developed to provide a basic capability, it does
not provide every feature that could possibly be desired for evaluating
netting concepts. There are obviously some additional models and algo-
rithms that could make TASRAN even more useful as an analysis tool.
Furthermore, the experience developed in running TASRAN during the test-
ing phase has led to some recommendations for improving its running
time. Some of the recommendations for additional models and for
improvements are summarized below.

4.1   AIRBORNE SURVEILLANCE RADARS

TASRAN at present includes only phased-array and rotating surveil-
lance radars fixed to the ground. Since AWACS and possible AWACS suc-
cessors may play a vital role in tactical air surveillance, the capabi-
lity to incorporate them into the TASRAN netting simulation should prove
extremely useful.

It is probably not essential, however, in modeling airborne sur-
veillance radars, to provide a very great level of detail. The critical
aspects of AWACS data in the netting system are those of knowing its
position and velocity; of correlating its tracks with those of other
sensors; and of communicating the tracking information to a common site.
The essential features of the models could be limited to acquiring and
tracking objects and reporting the tracks without attempting to provide
high-fidelity models of the radar's antenna pattern or its signal pro-
cessing system; other simulations could provide that level of detail.
Generic models, comparable in complexity to those provided for ground-
based radars in TASRAN, should meet the requirements for an airborne
surveillance radar in most netting analyses.

Once the basic models for an airborne radar are provided, any num-
ber of either phased-array or rotating surveillance radars of various
types could be easily simulated. The basic model for a moving radar

46

could be extended to radars on moving ground vehicles or even ships if desired.

4.2   SELF-SURVEY

One of the basic problems in a tactical air surveillance network is surveying-in the various sites in the net, particularly in wartime when sites are moving from time to time. TASRAN presently incorporates models for investigating the effects of surveying errors in the locations of the radar sites. It provides for both errors in horizontal position on the surface of the earth and errors in vertical and north angular alignment.

A netted system provides the capability to self-survey by means of common tracking of targets between surveyed and un-surveyed sites. The surveyed sites can report their estimates of the target position, and this target position can be used to deduce the location of the un-surveyed tracking site.

TASRAN is a nearly ideal tool for studying techniques for self-survey within an air surveillance network. The algorithms for self-survey can be built into TASRAN, and the TASRAN communication system can be modified to include new types of track reporting messages.

4.3   TARGET IDENTIFICATION

Identification of tactical aircraft may in the future include the combination of measurements and tracks from remote sensors along with transponder IFF. TASRAN could provide a testbed to analyze such techniques since it already incorporates a communication model which can pass different types of messages from site to site; since it models measurements and tracking; since it models configurations of many different types of sensors; and since it provides a scenario of many types of aircraft flying arbitrary flight paths.

47

4.4 IMPROVED TASRAN RUNNING TIME

Depending upon the complexity of the scenario and the netting sys-
tem to be anal_ ed, TASRAN can run from one to many times more slowly
than real time. That is, a minute of simulation time may require only
one minute of computer time for a simple scenario, but many times that
for a more complex scenario. If the user gets only a fraction of the
computer's resources, however, the amount of waiting time at a user's
terminal may grow excessive.

One means for speeding up TASRAN running time is to simplify some
of the radar and radar data processing models. Since TASRAN is modular,
some modules could be provided in both the current level of detail and a
simpler, faster version. For example, the CDT radar currently is simu-
lated pulse-by-pulse. However, it may be useful in some types of net-
ting analysis not to bother with the pulse-by-pulse detail of the radar.
The model might only provide periodic updates on those targets that
could be tracked by the radar. Thus the radar and data processing
modules could look at the actual positions of the targets within radar
coverage and generate track messages based on the actual target posi-
tion. These messages could be degraded with random errors, if desired,
without actually simulating radar measurements or the processing of
them.

Some of the modules in TASRAN where such simplification to improve
running time may be appropriate are the CDT and TWS radar modules and
the CDT and TWS data processing modules. The TASRAN communication sys-
tem already includes a simple, fast-running version. However, it could
be slightly upgraded to include some queuing and transmission delays now
available only in the two more slowly running versions. Most of the
other modules have little effect on TASRAN running time.

48
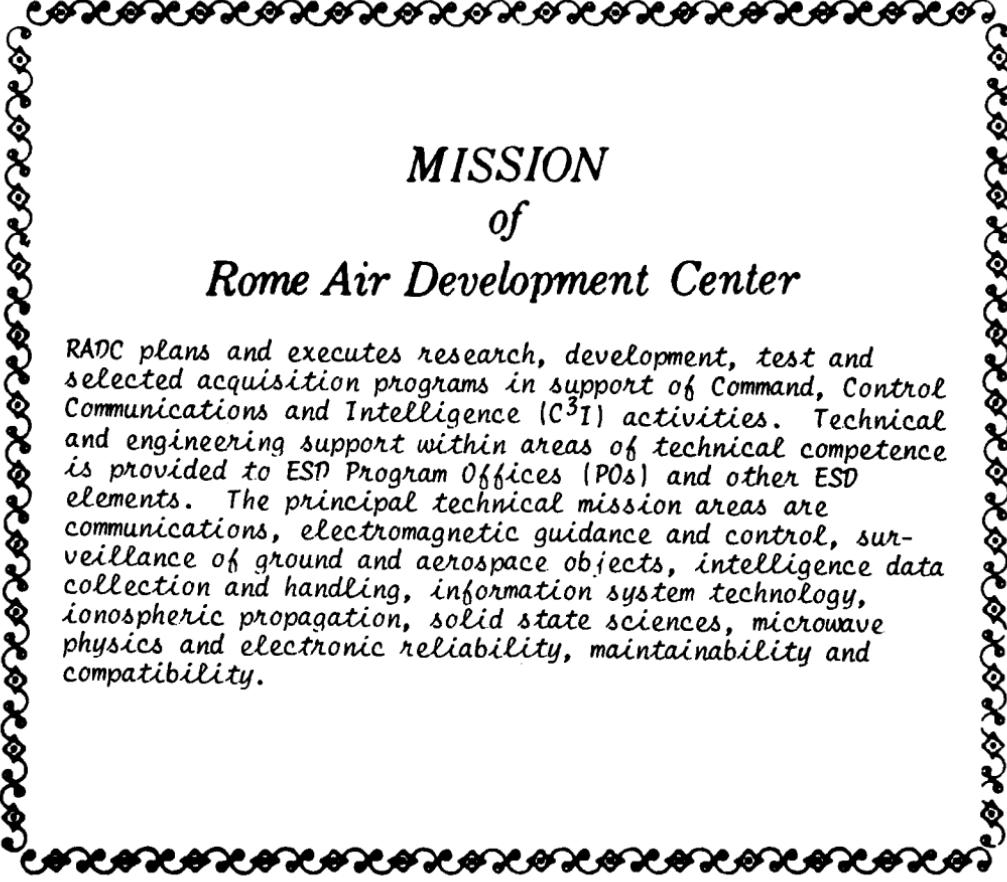
MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

DAT FILM